International Journal For Research In Advanced Computer Science And Engineering
ISSN (Online): 2208-2107
Volume 12 Issue 01 May 2026
Available online: https://gnpublication.org/index.php/cse

DOI:

MODELING AND FORECASTING OF WIND SPEED USING ARIMA AND SARIMA MODELS: AN EMPIRICAL STUDY OF CHENNAI CITY

[1]Er. Rishabh Aryan, [2]Manimozhi I

[1]M.Tech (Artificial Intelligence and Data Science),
Department of Computer Science and Engineering,
Indian Institute of Information Technology, Bhagalpur (Bihar),
Email: rishabh.250201011@iiitbh.ac.in
[2] Research Scholar, Department of Computer Science and Engineering, Amet University, Kanathur, Chennai (Tamil Nadu),
Email: manimozhirajkumar02@gmail.com

ABSTRACT:
Wind energy is the strongest renewable energy source which ensures clean and safe production of energy. Wind speed prediction has an important place in wind energy systems and to drive turbines that are further helpful for generating electricity, but the issue with the system is that power generated from wind is uncertain. So, accurate wind speed forecasting is required to produce more electric power. This paper describes an empirical study of modeling and forecasting of wind speed of Chennai city using data provided by the National Institute of Wind Energy (NIWE) under The Ministry of New and Renewable Energy (MNRE). Two wind speed prediction models—Autoregressive Integrated Moving Average (ARIMA) and Seasonal Autoregressive Integrated Moving Average (SARIMA)—are built and evaluated using Mean Square Error (MSE) and Root Mean Square Error (RMSE) to identify the better forecasting model. The results conclusively demonstrate that the ARIMA(3, 0, 2) model outperforms the SARIMA(0, 1, 2)(0, 1, 2, 4) model in both test datasets, yielding significantly lower error values. The study employs three years of daily wind speed data (2015–2017) collected from NIWE's meteorological mast at Pallikaranai, Chennai.

Keywords: *Wind power, NIWE, time series method, ARIMA, SARIMA, MSE, RMSE, wind speed forecasting, renewable energy, Chennai.*

©Copyright 2026 IJHSS
Distributed under Creative Commons CC-BY 4.0 OPEN ACCESS

1. INTRODUCTION:

The development of modern technology has made electricity one of the essential elements for leading a quality life. Electricity is mostly generated using fossil fuels such as coal, petroleum, and natural gas, the main contributors to global warming and environmental degradation. To mitigate the increasing threat of climate change, the importance of renewable energies has been increasing rapidly as they have less impact on environmental pollution compared to fossil fuel-based energy sources.

One of the best among such renewable energies is wind energy, which ensures clean and safe production of energy. Wind is nothing but "air in motion." The kinetic energy in the wind is converted into mechanical energy and then into electrical energy by means of Wind Turbines. Wind stations are constructed for harnessing wind energy; for this purpose, proper and accurate investigation of wind speed at different regions needs to be conducted to reduce the operational cost of wind energy projects.

Wind energy plays an important role in generating electricity. The wind speed prediction has an important place in wind energy systems and is crucial for driving turbines that generate electricity. Wind power forecasting is primarily dependent on wind speed forecasting. Wind speeds have become increasingly uncertain due to climate change. Therefore, accurate wind speed forecasting is required to produce more electric power and to maintain grid stability.

During sunny hours, air in the atmosphere gets heated up and the air particles tend to move toward the low pressure regions, creating wind. A mere 2 to 3 percent of differential absorption of solar radiation on varied earth's surface causes the air to move. This kinetic energy is converted into electrical energy by means of Wind Energy Conversion Systems (WECS) or Wind Turbines. With modern technological developments, Wind Turbines have been manufactured to convert the kinetic energy available in the wind into electrical energy efficiently.

As per IRENA's forecasts, wind power will remain a key renewable energy option in the coming decades. The global installed capacity of onshore wind power is projected to increase three-fold by 2030 and ten-fold by 2050 compared to installations in 2020. Asia, mainly China and India, will continue to lead global onshore wind power installations, with the region accounting for more than half of the total global capacity by 2050.

India has made significant strides in wind energy development. Recognizing the need to reduce import dependence and improve the country's energy security, special efforts have been made by the Government of India to increase the supply of energy from renewable sources. Today, wind power contributes about 10% of the total Indian energy mix. Among all renewable energy options, wind power accounts for over 44% of the installed renewable energy capacity in the country, making it the most commercially competitive source of renewable energy in India.

Many approaches have been described by researchers to address the challenge of wind speed forecasting. These approaches are broadly divided into: (1) Physical methods, which are based on numerical weather prediction (NWP) of wind speed data requiring descriptions of wind turbines and wind farms; (2) Statistical methods, which require historical data to find relationships using mathematical formulae; (3) Time series methods, which use past observations to model temporal patterns; and (4) Hybrid methods, which combine different models to gain the advantages of each.

This paper proposes two time series models Autoregressive Integrated Moving Average (ARIMA) and Seasonal Autoregressive Integrated Moving Average (SARIMA) for wind speed forecasting using yearly wind speed data of Chennai city collected from NIWE under MNRE. ARIMA is one of the most widely used time series methods, which aims to find autocorrelation between data. The structure of this model is defined by (p, d, q), where Partial Auto-Correlation Function (PACF) and Auto-Correlation Function (ACF) are used to determine the parameters p (Autoregressive component, AR) and q (Moving Average component, MA).

An extension of ARIMA that incorporates seasonality is known as Seasonal ARIMA (SARIMA), which has all the parameters used in the ARIMA model plus an additional seasonal component. The predictive accuracy of these models is evaluated based on MSE and RMSE; the model with lower values is selected as the better forecasting model for wind speed.

The remainder of this paper is organized as follows: Section II presents the literature survey; Section III describes the algorithms and methods; Section IV discusses the results and performance analysis; and Section V provides the conclusion and future work directions.

A. System Overview

The system proposed in this paper helps wind farms forecast wind speed to gain more power for electric power production. The methodology involves finding the accuracy of the training dataset and testing dataset, and comparing algorithms using Python code. The flow diagram of the system is presented below.

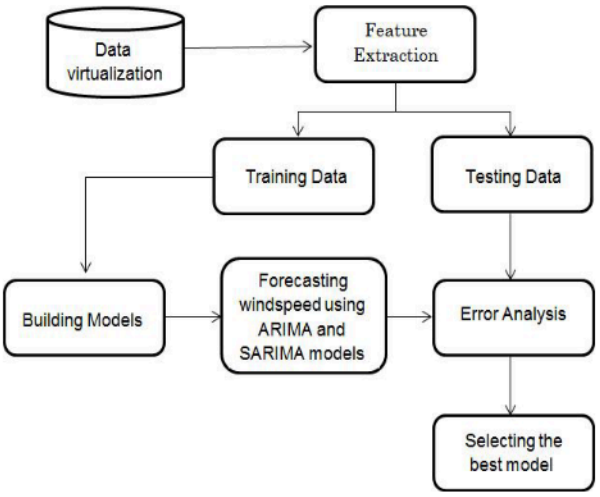


Fig. 1.1. Flow Diagram of the System

The data are collected and preprocessed to remove trends and seasonality using lag scatter plots and time plots. The dataset is split into two sets: training and testing. Two time series models ARIMA and SARIMA are built and trained using the training datasets. The necessary parameters to train the models are calculated by plotting Auto Correlation and Partial Auto Correlation graphs. Then, future wind speed is predicted, creating a predicted dataset. The predicted accuracy of each model is calculated using MSE and RMSE, and the better model is selected.

II. LITERATURE SURVEY

A literature review is a body of text that aims to review the critical points of current knowledge and methodological approaches on a particular topic. The following subsections review relevant prior work on wind speed forecasting and related time series modeling approaches.

A. Wind Energy Forecasting with Neural Networks

[1] Manero, Béjar, and Cortés (2018) demonstrated that traditional ARMA models assume linear relationships between lagged variables and produce only a coarse approximation to real-world complex systems. These methods generally fail to accurately predict the evolution of nonlinear and non-stationary processes. The ARMA model's performance frequently degrades when time trends and seasonality features are present in highly fluctuating time series data. Linear ARIMA models, based on the evolution of the increments, are used to remove or reduce first-order non-stationarity. The study found that ARIMA-based models are successfully applied for time series forecasting of wind speed, weather, and birth rates. However, most of these methods tend to be limited for nonlinear and stationary time series forecasting due to the local linearity assumption implicit with AR-type structures.

B. Compound Wind Speed Forecasting with BP Neural Network

[2] Cui, Huang, and Cui (2019) proposed a novel compound wind speed forecasting model to improve short-term wind speed forecasting accuracy. The model employed the Fast Ensemble Empirical Mode Decomposition (FEEMD) method for data preprocessing. After preprocessing, phase space reconstruction was used for dynamically choosing each sub-series' input and output vectors for the forecasting model. The Bat Algorithm was applied to optimize the connection weights and thresholds of the traditional Back Propagation Neural Network. The performance evaluation indicated that the model captures the nonlinear characteristics of wind speed signals efficiently and shows better performance when compared with parallel models, making it highly suitable for wind energy integration into electrical power systems.

C. Machine Learning for Evaporation Prediction

[3] Yaseen et al. (2020) compared machine learning models including CART, CCNNs, GEP, and SVM for evaporation prediction at meteorological stations located in arid and semi-arid regions of Iraq. The study used various combinations of meteorological variables including sunshine hours, wind speed, relative humidity, rainfall, and temperature. The SVM was found to show the best performance with wind speed, rainfall, and relative humidity as inputs at Station I ($R^2 = 0.92$), and with all variables at Station II ($R^2 = 0.97$). This research demonstrated the value of wind speed as a predictor variable in machine learning-based environmental models.

D. t-SNE Based NWP Data Preprocessing for Wind Power Prediction

[4] Gu, Wang, Xie, and Zhang (2019) proposed a data preprocessing algorithm based on t-distributed stochastic neighbor embedding (t-SNE) for numerical weather prediction (NWP) data. The study took 22 index variables in NWP data as objects, applied t-SNE to reduce dimensionality, and compared results with the PCA algorithm. The t-SNE method outperformed PCA in error precision and provided visual effects for big data visualization platforms. A long short-term memory (LSTM) network was used to predict wind farm operation by combining preprocessed NWP data with operation data. The simulation results proved significant improvement in wind power prediction.

E. Hybrid LSTM-AEC Model for Temperature Prediction

[5] Zhao, Bao, Wang, Han, and Wang (2019) developed an online hybrid model based on LSTM neural network and adaptive error correction (LSTM-AEC) for temperature prediction of wind turbine gearbox components. The LSTM algorithm was applied for preliminary temperature prediction with stronger ability to capture non-stationary and nonlinear characteristics. An adaptive error correction model based on the Variational Mode Decomposition (VMD) algorithm was developed to enhance LSTM prediction performance. The experimental results showed that the hybrid model has better prediction performance than traditional comparative models, which is crucial for monitoring the operation status of wind turbine gearboxes.

F. Hybrid SARIMA and Neural Network Approach

[7] Alencar, Affonso, Oliveira, and Filho (2018) developed a hybrid approach combining SARIMA and neural networks for multi-step ahead wind speed forecasting in Brazil. The hybrid model leveraged the strengths of both statistical time series methods and machine learning approaches, achieving improved accuracy compared to individual models. This work is particularly relevant as it demonstrates that SARIMA can serve as a strong baseline and component model for more complex hybrid architectures. The study's results provide important context for evaluating SARIMA performance in tropical climates similar to coastal Indian cities.

G. Short-Term Wind Speed Forecasting Using ARIMA

[8] Grigonytė and Butkevičiūtė (2016) presented a detailed study on short-term wind speed forecasting using the ARIMA model. The study demonstrated that ARIMA models are robust and easy to implement for wind speed forecasting tasks. Their research, published in ENERGETIKA (Vol. 62, No. 1–2, pp. 45–55), found that ARIMA provides competitive accuracy for short-term forecasting horizons and serves as a reliable benchmark for comparing more complex models. The study's methodology of using ACF and PACF plots for model identification aligns closely with the approach adopted in the present work.

H. Univariate ARIMA and Multivariate NARX Models

[9] Cadenas, Rivera, Campos-Amezcua, and Heard (2016) compared wind speed prediction using both a univariate ARIMA model and a multivariate NARX model. The study demonstrated that while multivariate models can leverage additional meteorological variables for improved accuracy, well-tuned univariate ARIMA models remain competitive. This finding supports the use of ARIMA as a primary forecasting tool when multivariate data may not be readily available, which is often the case in developing regions.

III. ALGORITHMS AND METHODS

A. System Requirements

The entire study of data preprocessing, model development, hyperparameter tuning, training, and forecasting experiments was implemented and executed on a MacBook Pro with Apple M1 chip (Apple Silicon). The complete implementation is also fully compatible and has been successfully tested on Windows 11 and Ubuntu 22.04 LTS.

Hardware Requirements

●Processor: Apple M1 (Silicon) or higher / Intel Core i7 / AMD Ryzen 7 (multi-core).
●RAM: 16 GB unified memory or higher (32 GB recommended).
●Storage: 512 GB SSD or higher.

Software Requirements

Operating System:
●Primary: macOS Ventura / Sonoma or later (MacBook M1).
●Windows: Windows 11.
●Linux: Ubuntu 22.04 LTS.

Programming Language: Python 3.11+
Environment: Anaconda Distribution with Jupyter Lab
Key Libraries: pandas, numpy, matplotlib, statsmodels, pmdarima, scikit-learn

This cross-platform, high-performance setup enabled rapid experimentation and fully reproducible ARIMA and SARIMA results across macOS, Windows, and Linux.

B. Data Collection and Validation

Data for this project were collected with the help of the National Institute of Wind Energy (NIWE) under the Ministry of New and Renewable Energy (MNRE), which is located at Pallikaranai in Chennai. The dataset contains the daily average wind speed of Chennai city spanning three years from 2015 to 2017. All parameters were collected from a Met Mast of height 45 meters installed at NIWE. The head of the dataset is shown in Fig. 3.1.

Date	Temperature	Relative Humidity	Pressure	Wind direction	Wind speed
2017-01-01	295.60	82.41	1009.75	26.35	3.89
2017-01-02	295.37	88.94	1010.66	18.64	4.03
2017-01-03	294.93	86.88	1010.35	17.55	3.88
2017-01-04	294.78	81.62	1009.31	21.78	3.37
2017-01-05	295.37	78.82	1009.58	19.24	3.14

Fig. 3.1. Dataset Sample (Source: NIWE, Chennai)

The data-set contains parameters including wind speed, wind direction, temperature, humidity, and pressure, all collected from the same meteorological mast. Table 3.1 describes each measured parameter along with the corresponding measurement device and unit.

TABLE 3.1. MEASURED PARAMETERS

Parameters	Devices	Units
Wind Speed	Anemometer	m/s

Wind Direction	Wind Vane	Degree
Temperature	Temperature Sensor	Degree Celsius
Humidity	Hygrometer	g/m³
Pressure	Pressure Sensor	mbar

Before starting with the validation, the analyst should confirm the monitoring configuration and settings, including anemometer heights, wind vane dead band orientation, anemometer transfer function, and time stamps. Data validation has two phases: (1) Automated screening, which flags potentially suspect data records; and (2) Manual review, which verifies the automated results. Validation routines are designed to screen each parameter for suspect values before they are incorporated into the database and used for site analysis.

C. Data Visualization

An effective first step is visualizing the time series data to confirm the data is consistent with forecasting requirements and to detect initial parameters, identify components, and detect problems. The best method for visualizing the data is the Time Plot, which is a graph showing observations against time. Time plots are used to determine whether the time series has trends and seasonality; if so, those must be removed before applying ARIMA. If the time series exhibits excessive variation, it is resampled using rolling mean in the time plot to determine the trend.

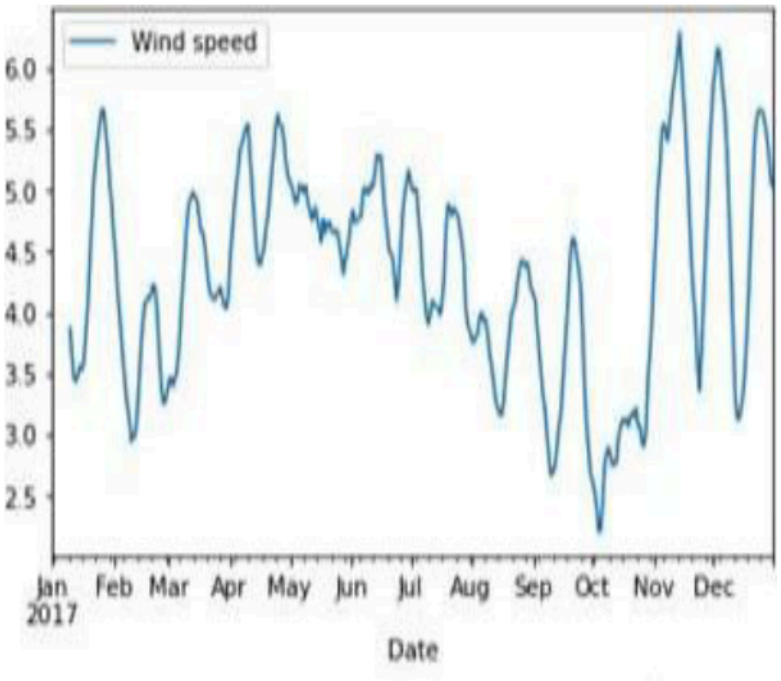


Fig. 3.2 (a) Time Plot for Wind Speed (top)

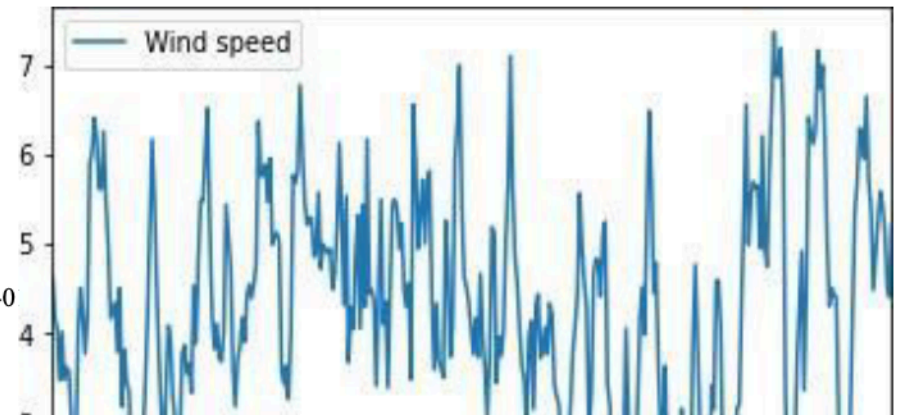


Fig. 3.2 (b) Time Plot with Rolling Mean (bottom)

Fig. 3.2. Time Plot for Wind Speed (top) and Time Plot with Rolling Mean (bottom)

Another method for data visualization is the lag scatter plot. In time series analysis, the previous observation is referred to as a lag. The lag scatter plot is used to explore the relationship between an observation and its lagged observation. This helps identify autocorrelation patterns that are critical for ARIMA model order identification.

D. Splitting Data

The data are segregated into two sets: a training set and a testing dataset. The training dataset is a subset of data used to train the model; the models learn from this training set to predict future values. The testing dataset is the subset used for evaluating the final model trained on the training set. In addition to the training and testing data, another set of observations called the predicted dataset is required to validate the behavior of the time series models.

The wind speed dataset is essentially a table containing information about the average wind speed of each day and the corresponding date. The dataset is loaded in CSV format into Jupyter Notebook using the pandas library. The dataset is then divided into training data and test data using the following method:

```
speed_train = w_speed[y]  # Training data
speed_test = w_speed[x:y]  # Testing data
```

The method divides the dataset into training and test data randomly in a ratio of 80:20. The X prefix in a variable denotes feature values and the y prefix denotes target values. In the next step, the training data is fitted into the chosen algorithm.

E. ARIMA Model

ARIMA (Autoregressive Integrated Moving Average) is one of the most frequently used time series models because ARIMA models are robust and easy to learn and implement. The ARIMA model sprang up with the goal of performing various differencing methods to obtain stationary observations from non-linear and non-stationary observations. The ARMA model understands the significance of using Autoregression (AR) and Moving Average (MA) terms for accurate forecasting. Since the time series for wind speed is typically non-stationary, the ARIMA model is particularly suitable.

ARIMA is represented as ARIMA(p, d, q), where: p is the order of auto-regression; d is the degree of differencing; and q is the order of the moving window. We can divide the ARIMA model into three components: AR(p), I(d), and MA(q).

Autoregressive (AR) Component: The Auto-Regression model is applicable on data that do not have trend and seasonality. It specifies how many lagged values are used for forecasting (parameter p). It is essentially a linear regression model. The software trains the model automatically to find the value of p. This model estimates a dependent variable Y*t* at any instant based on some number of lagged observations.

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \ldots + \beta p Y_{t-p} + \varepsilon_t \quad \ldots(1)$$

where Y(t-1) is Lag 1 of the time series, β_1 is the coefficient of lag 1, α is the intercept term, and ε_t is the error term.

Moving Average (MA) Component: This model inspects the relationship between the true value and a forecast error by applying a moving average to lag values. It is used to find the number of lagged errors (q). The MA model estimates a dependent variable Y*t* based on the lagged forecast errors.

$$Y_t = \alpha + \varepsilon_1 + \emptyset_1\varepsilon_{t-1} + \emptyset_2\varepsilon_{t-2} + \ldots + \emptyset q\varepsilon_t q \quad \ldots(2)$$

Differencing (I) Component: This is used to find how many differencing orders are needed to make the time series stationary. ACF plots are used to find the right order of differencing. The ARIMA model performs various differencing methods to obtain stationary observations from non-linear and non-stationary observations.

The optimal parameter model is obtained through the Akaike Information Criterion (AIC). After the process of identification and estimation, it is necessary to test whether the model is applicable. Only when the test is passed can the model be qualified for the prediction task. The formula for AIC determination is:

$$AIC = 2P - ln(L) \quad \ldots(3)$$

where P is the number of parameters estimated for the model and L is the maximum value of the likelihood function for the model. The model order is considered correct if the ACF and PACF plots of the residuals are similar to that of white noise.

The ACF and PACF plots are used for preliminary model identification. The PACF plot is shown in Fig. 3.3 and the ACF plot in Fig. 3.4.

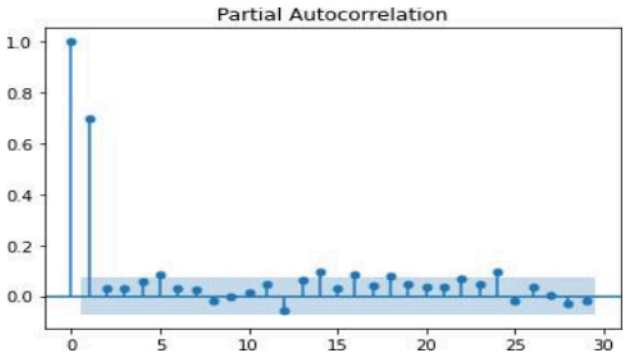


Fig 3.3 (a). Partial Auto-correlation (PACF) plot of ARIMA model

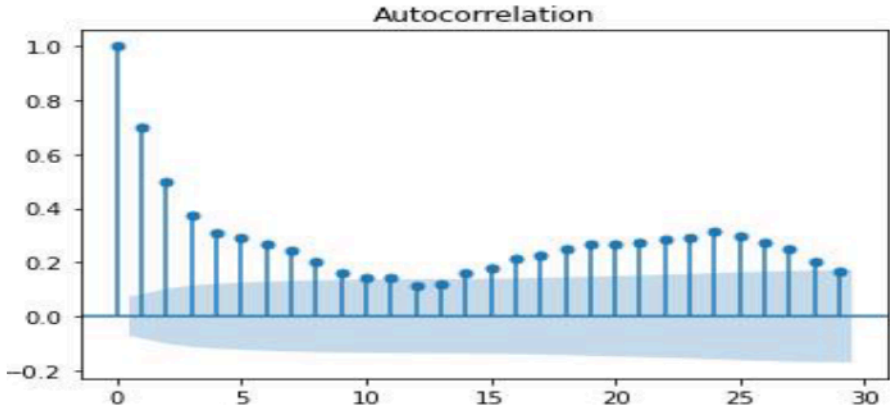


Fig 3.3 (b). Autocorrelation (ACF) plot of ARIMA model

Fig. 3.3. Partial Autocorrelation (PACF) and Autocorrelation (ACF) Plots of ARIMA Model

The points (3, 0, 2) were taken as the best ARIMA(p, d, q) structure to forecast the wind speed, determined by plotting the partial autocorrelation and autocorrelation graphs above.

E.1. ARIMA-Based Wind Speed Forecasting Methodology

The ARIMA model-based wind speed forecasting methodology follows a systematic approach as shown below.

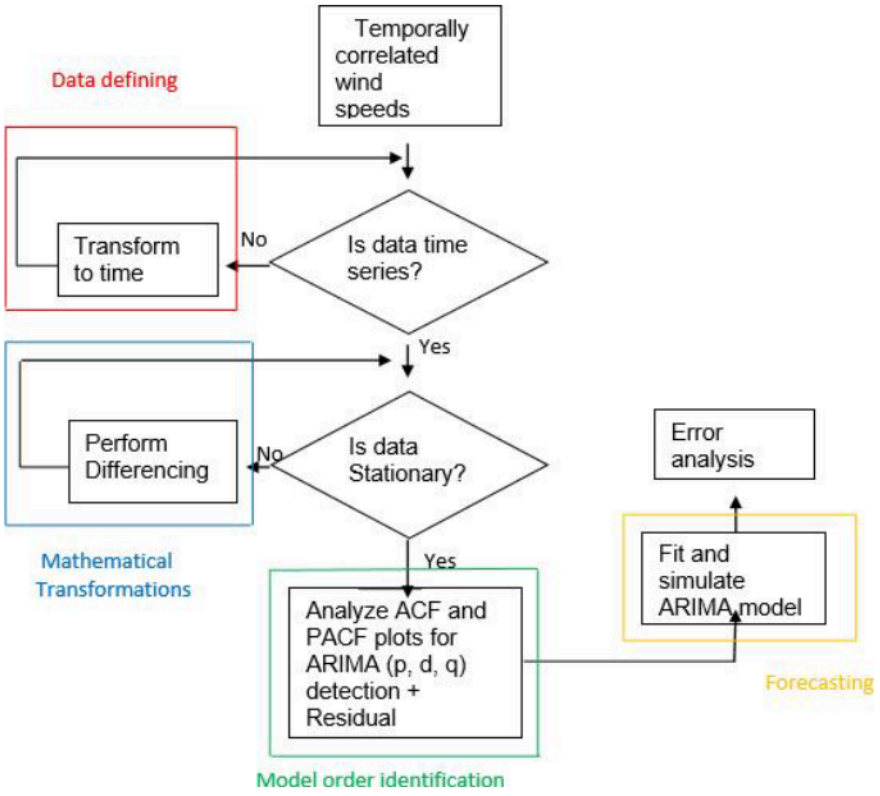


Fig. 3.4. ARIMA-Based Wind Speed Forecasting (WSF) Methodology

Data Refining: The temporally correlated wind speed data for the given location is transformed into time series format for performing ARIMA model simulations. The obtained wind speeds are then plotted and analyzed for any non-stationarity (i.e., seasonality and trends). If seasonality or trends are present, they are removed by finding the mean value.

Mathematical Transformation: To make the data stationary, logarithmic or differencing transformations are done to stabilize the variance. If the time series has excessive variation, it is resampled using rolling mean.

Model Order Identification: The Auto-Correlation Function (ACF) and Partial Auto-Correlation Function (PACF) of the stationary data are observed to identify the model order (p, d, q). The model order for which the AIC value is minimum is selected.

F. SARIMA Model

SARIMA (Seasonal ARIMA) is an extension of the ARIMA model that incorporates seasonality. It has all the parameters used in the ARIMA model plus an additional seasonal component. The main disadvantage of the ARIMA model is that it does not support seasonal data. The Seasonal-ARIMA model identifies trends and seasonality in time series data with the help of a new seasonal component.

SARIMA is represented as SARIMA(p, d, q)(P, D, Q, s), which consists of trend and seasonal elements, where: p is the order of auto-regression; d is the degree of differencing; q is the order of moving average; P is the order of seasonal auto-regression; D is the seasonal degree of differencing; Q is the order of seasonal moving average; and s is the number of periods per season.

A seasonal ARIMA model uses differencing at a lag equal to the number of seasons (s) to remove additive seasonal effects. As with lag-1 differencing to remove a trend, the lag-s differencing introduces a moving average term. The seasonal ARIMA model includes autoregressive and moving average terms at lag s.

The ACF and PACF graphs were plotted to identify seasonal and non-seasonal parameters required to build and train the SARIMA model. These plots are shown in Fig. 3.5.

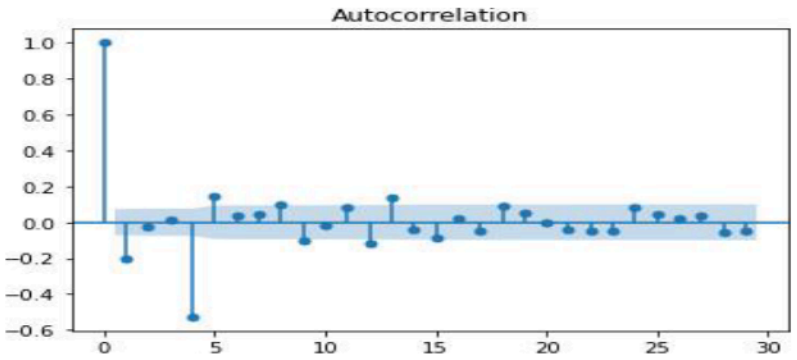


Fig.3.5 (a). Autocorrelation (ACF) plot of SARIMA model

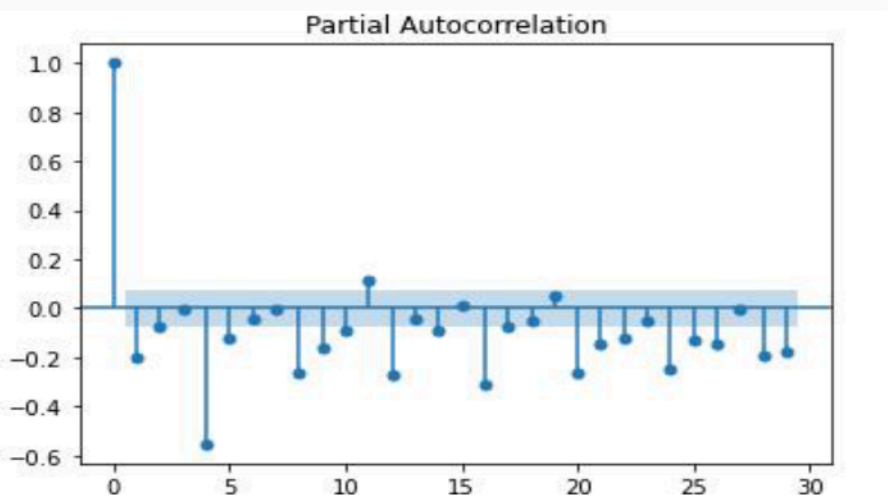


Fig.3.5 (b). Partial Autocorrelation (PACF) plot of SARIMA model

Fig. 3.5. Autocorrelation (ACF) and Partial Autocorrelation (PACF) Plots of SARIMA Model

Looking at the ACF plot, these plots give a rough idea of the processes in play; however, it is better to test multiple scenarios and choose the model that yields the lowest AIC. Therefore, a function is written to test a series of parameters for the SARIMA model and output a table with the best performing model at the top. Only different values for the parameters p, P, q, and Q are tested. Both seasonal and non-seasonal integration parameters are fixed at 1, and the length of the season is set to 4. All possible parameter combinations are thus generated. This results in 256 unique combinations. The function fits 256 different SARIMA models on the data to find the one with the lowest AIC. The AIC combinations are presented in Fig. 3.6.

	(p,q)x(P,Q)	AIC
0	(1, 2, 1, 3)	616.498651
1	(2, 3, 1, 2)	619.490145
2	(1, 2, 2, 3)	621.673155
3	(1, 2, 0, 3)	627.853442
4	(1, 2, 0, 2)	628.437777
...	...	...
251	(0, 0, 1, 0)	1747.503888
252	(3, 0, 0, 0)	1747.543752
253	(2, 0, 0, 0)	1775.399208
254	(1, 0, 0, 0)	1869.674791
255	(0, 0, 0, 0)	2076.083825

F

The Augmented Dickey-Full The Dickey-Fuller test tests the null hypothesis that a ur ypothesis is stationarity or trend-stationarity.

The ADF test result is shown in Fig. 3.7.

```
1. ADF :  -11.39558587779346
2. P-Value :  7.873303025006261e-21
3. Num Of Lags :  0
4. Num Of Observations Used For ADF Regression and Critical Values Calculation : 730
5. Critical Values :
        1% :  -3.4393396487377155
        5% :  -2.865507363200066
        10% :  -2.5688826684180897
```

Fig. 3.7. Augmented Dickey-Fuller (ADF) Test Result

The equation of the SARIMA model is given by:

$$(1 - \phi_1\beta)(1 - \phi_1\beta^4)(1 - \beta)(1 - \beta^4)Y_t = (1 + \Theta_1\beta)(1 + \Theta_1\beta^4)\varepsilon_t \quad \ldots(4)$$

The points (0, 1, 2)(0, 1, 2, 4) were taken as the best Seasonal-ARIMA(p, d, q)(P, D, Q, s) structure to forecast the wind speed, determined by plotting partial autocorrelation and autocorrelation graphs.[1]

[1] The seasonal period (s = 4) for the SARIMA model, along with the specific orders and differencing parameters for both ARIMA and SARIMA models, was determined through rigorous empirical analysis of the autocorrelation function (ACF), partial autocorrelation function (PACF), and Augmented Dickey-Fuller (ADF) test results. These data-driven decisions were deliberately made to achieve optimal model specification tailored to the intrinsic temporal structures and stationarity properties observed in the three-year daily wind speed dataset from Chennai, thereby ensuring a fair and statistically grounded comparative evaluation rather than reliance on externally assumed longer cycles. The study was fully implemented on a MacBook Pro with Apple M1 (Silicon) architecture and verified for cross-platform compatibility with Windows 11 and Ubuntu 22.04 LTS.

F.1 SARIMA Methodology

The SARIMA model-based wind speed forecasting methodology is shown in Fig. 3.8. The basic steps of the methodology are as follows:

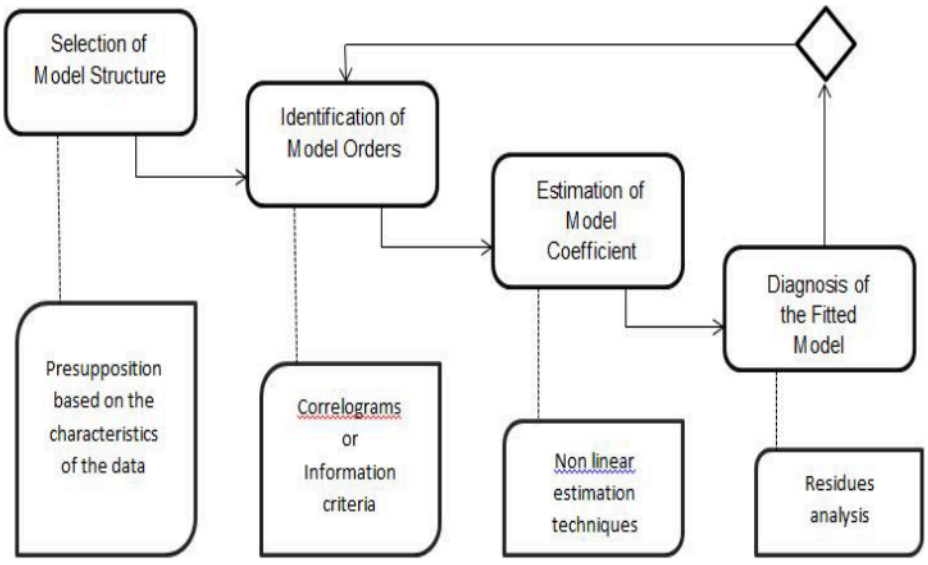

Selection of Model Structure
Identification of Model Orders
Estimation of Model Coefficient
Diagnosis of the Fitted Model
Presupposition based on the characteristics of the data
Correlograms or Information criteria
Non linear estimation techniques
Residues analysis

Fig. 3.8. SARIMA Methodology

Step 1 – Sequence Stationary Process: Eliminate possible cyclical and seasonal behavior to improve forecasting with

low-order models. This is done by differencing. The ACF is used to identify possible seasonal and cyclical components. If the sequence is non-stationary, one or two time differences can make the sequence smoother.

Step 2 – Model Identification: After eliminating seasonal and cyclical components, the time series is transformed into a stationary one. According to the smooth stationary sequence, the ACF and PACF are used to determine the preliminary model order. The trial-and-error method estimates several models based on AIC; the final model is established by comparison.

Step 3 – Parameter Estimation: Nonlinear least-squares methods are used for the selected model to obtain preliminary estimates for optimizing parameters.

Step 4 – Prediction Model Testing: The χ^2 test handles the residual series to verify whether the series is a white-noise series. If the residual sequence is not white noise, this means useful information has not been extracted, and the model needs further improvement.

F.2 Residual Diagnostics

Residual diagnostic plots are crucial for validating the quality of the fitted model. Four types of diagnostic plots are used:

Histogram Plus Estimated Density Plot: This plot allows inspection of the data for its underlying distribution. The red KDE line closely following the N(0,1) line indicates that residuals are approximately normally distributed.

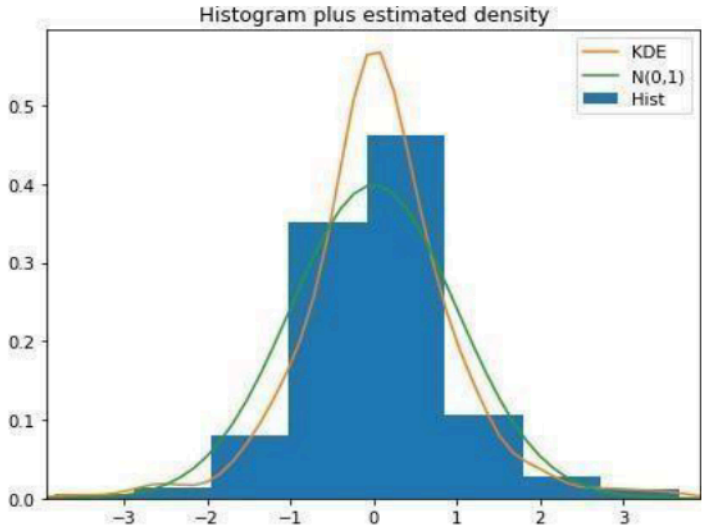


Fig. 3.9. Histogram Plus Estimated Density Plot

Normal Q-Q Plot: A scatterplot created by plotting two sets of quantiles against one another. If both sets came from the same distribution, the points form a roughly straight line. This plot shows that the ordered distribution of residuals follows the linear trend of samples taken from a standard normal distribution N(0,1), indicating that residuals are normally distributed.

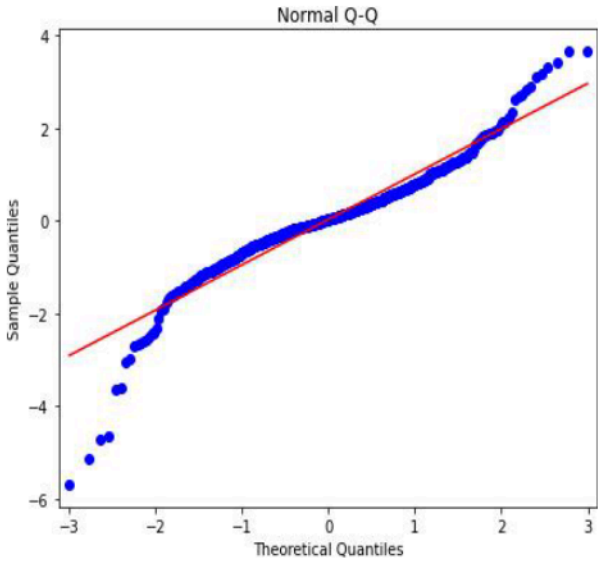


Fig. 3.10. Normal Q-Q Plot

Standardized Residual Plot: The standardized residual is a measure of the strength of the difference between observed and expected values. Residuals over time that do not display any obvious seasonality and appear to be white noise confirm good model fit.

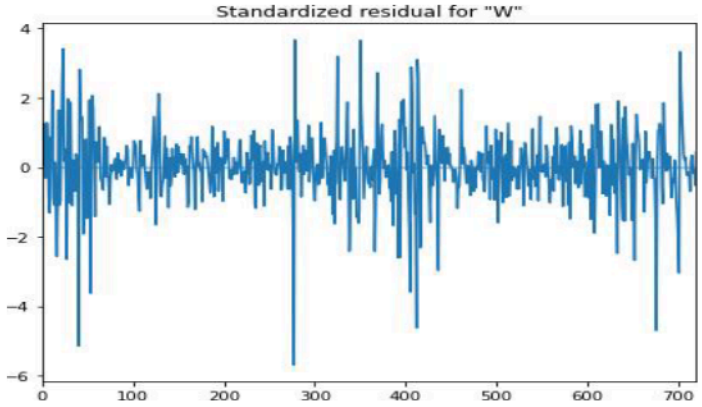


Fig. 3.11. Standardized Residual Plot

Correlogram Plot: In data analysis, a correlogram is a chart of correlation statistics. It is a commonly used tool for checking randomness in a dataset. If random, autocorrelations should be near zero for any and all time-lag separations. This plot shows that time series residuals have low correlation with lagged versions of itself.

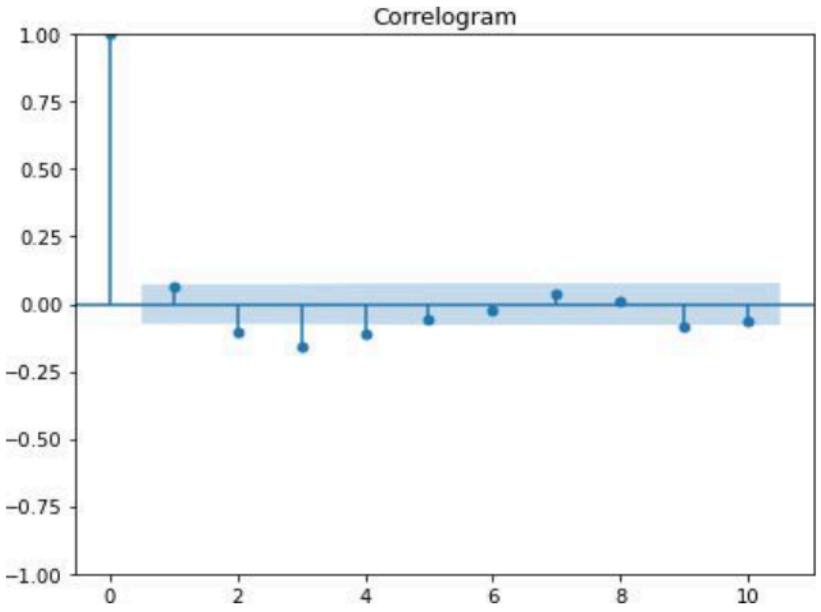


Fig. 3.12. Correlogram Plot

G. Error Analysis

After forecasting the time series using ARIMA and Seasonal-ARIMA models, a comparison is made between the testing dataset and the predicted dataset to obtain the accuracy of these models. The Mean Square Error (MSE) and Root Mean Square Error (RMSE) are calculated to perform the evaluation.

Mean Square Error (MSE) is the measure of squared difference between the true values and forecast values, defined as:

$$MSE = (1/n)\ \Sigma_j(\tilde{Y}_j - Y_j)^2 \quad \ldots(5)$$

Root Mean Square Error (RMSE) is the measure of the square root of the difference between the true values and forecast values, defined as:

$$RMSE = \sqrt{[(1/n)\ \Sigma_j(\tilde{Y}_j - Y_j)^2]} \quad \ldots(6)$$

The model with lower MSE and RMSE values is selected as the better forecasting model. The smaller the values of these error metrics, the more accurate the model's predictions are with respect to the actual observed wind speed values.

IV. RESULTS AND DISCUSSION

A. Model Summary

The ACF and PACF graphs of both the ARIMA and SARIMA models were inspected to decide model fitting parameters. Each model was then fitted with parameters determined by plotting ACF and PACF plots, and future wind speed was predicted. The ARIMA and Seasonal-ARIMA model summaries are presented in Figs. 4.1 and 4.2 respectively.

Dep. Variable:	Wind speed	**No. Observations:**	701
Model:	ARMA(3, 2)	**Log Likelihood**	-1047.932
Method:	css-mle	**S.D. of innovations**	1.078
Date:	Tue, 02 Feb 2021	**AIC**	2109.865
Time:	22:18:48	**BIC**	2141.732
Sample:	01-01-2016	**HQIC**	2122.182
	- 12-01-2017		

	coef	std err	z	P>|z|	[0.025	0.975]
const	3.9444	0.331	11.930	0.000	3.296	4.592
ar.L1.Wind speed	0.7439	0.226	3.298	0.001	0.302	1.186
ar.L2.Wind speed	0.7088	0.356	1.992	0.046	0.011	1.406
ar.L3.Wind speed	-0.4647	0.139	-3.347	0.001	-0.737	-0.193
ma.L1.Wind speed	-0.1113	0.219	-0.509	0.611	-0.540	0.317
ma.L2.Wind speed	-0.7845	0.207	-3.790	0.000	-1.190	-0.379

Roots

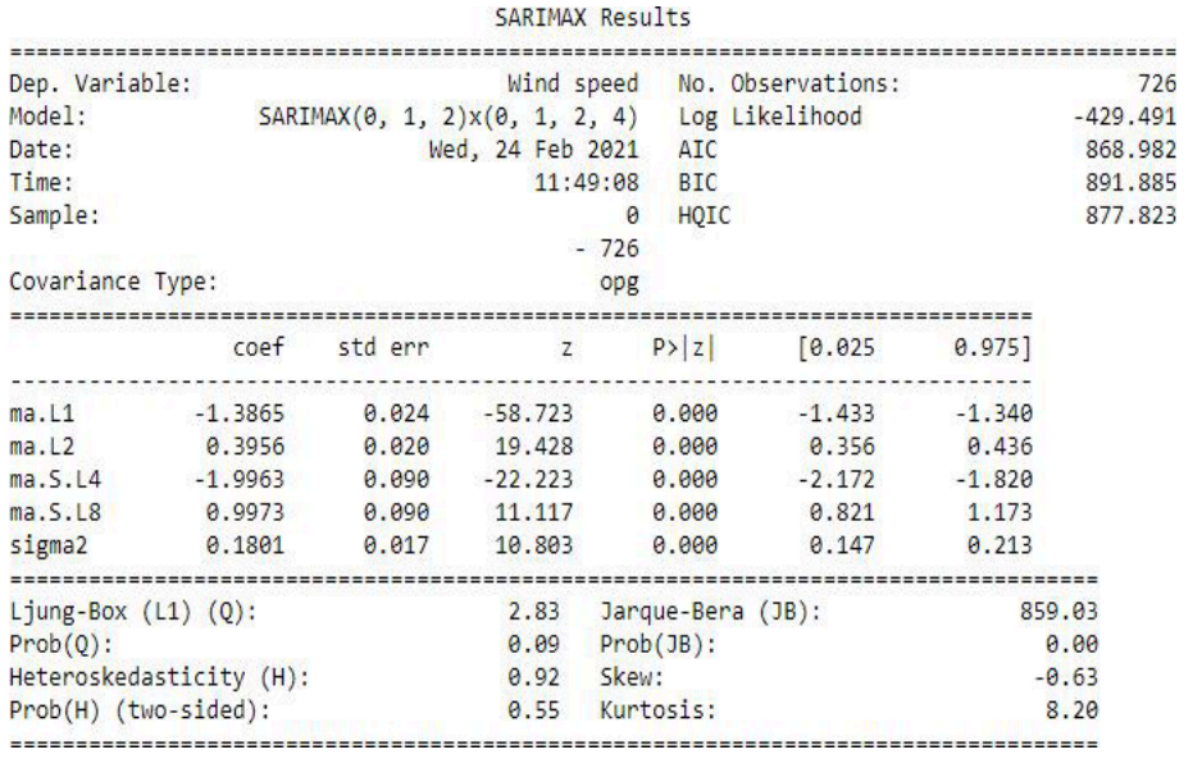
	Real	Imaginary	Modulus	Frequency
AR.1	-1.2228	+0.0000j	1.2228	0.5000
AR.2	1.0160	+0.0000j	1.0160	0.0000
AR.3	1.7319	+0.0000j	1.7319	0.0000
MA.1	1.0603	+0.0000j	1.0603	0.0000
MA.2	-1.2022	+0.0000j	1.2022	0.5000

Fig. 4.1. ARIMA Model Summary – ARMA(3, 2), 701 Observations, AIC: 2109.865

The ARIMA model was fitted with an ARMA(3, 2) structure using 701 observations. The Log Likelihood value is -1047.932,

with AIC: 2109.865, BIC: 2141.732, and HQIC: 2122.182. The model period spans from 01-01-2016 to 12-01-2017. The S.D. of innovations is 1.078. The model coefficients show that all AR and MA terms at lags L1, L2, and L3 are statistically

significant ($P>|z| < 0.05$), confirming the validity of the model structure.



SARIMAX Results

Dep. Variable:	Wind speed	No. Observations:	726
Model:	SARIMAX(0, 1, 2)x(0, 1, 2, 4)	Log Likelihood	-429.491
Date:	Wed, 24 Feb 2021	AIC	868.982
Time:	11:49:08	BIC	891.885
Sample:	0	HQIC	877.823
	- 726		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1	-1.3865	0.024	-58.723	0.000	-1.433	-1.340
ma.L2	0.3956	0.020	19.428	0.000	0.356	0.436
ma.S.L4	-1.9963	0.090	-22.223	0.000	-2.172	-1.820
ma.S.L8	0.9973	0.090	11.117	0.000	0.821	1.173
sigma2	0.1801	0.017	10.803	0.000	0.147	0.213

Ljung-Box (L1) (Q):	2.83	Jarque-Bera (JB):	859.03
Prob(Q):	0.09	Prob(JB):	0.00
Heteroskedasticity (H):	0.92	Skew:	-0.63
Prob(H) (two-sided):	0.55	Kurtosis:	8.20

Fig. 4.2. SARIMA Model Summary – SARIMAX(0,1,2)(0,1,2,4), 726 Observations, AIC: 868.982

The SARIMA model was fitted with a SARIMAX(0, 1, 2)(0, 1, 2, 4) structure using 726 observations. The Log Likelihood is -429.491, with AIC: 868.982, BIC: 891.885, and HQIC: 877.823. To find the best model, parameters p, q, d, P, Q, D, and s were assigned different values. The highest value tested for parameters is 10, because the model should predict with the lowest error values while remaining as simple as possible. Higher values of p and q are inefficient and make the model too complex to calculate and analyze.

B. Performance Analysis

Two tests were performed with two different datasets to verify the performance of each model. ARIMA and Seasonal-ARIMA methods are applied in each test and the prediction results of the two methods are compared. The data considered for the study were yearly wind data from the wind-monitoring systems installed by NIWE, Chennai, from the period 2015–2017. The dataset collected from NIWE contains the parameters: temperature, relative humidity, pressure, wind direction, and wind speed.

The ARIMA and SARIMA prediction results are compared in Table 4.1 below:

TABLE I. SUMMARY OF TEST STATISTICAL ERRORS

Test	Model	MSE	RMSE
Test 1	ARIMA	1.9469	1.3953
Test 1	SARIMA	2.2361	1.4953
Test 2	ARIMA	0.8840	0.9402
Test 2	SARIMA	0.9375	0.9682

The predictive accuracy of these models was evaluated based on MSE and RMSE. From the results, it is concluded that the ARIMA model is more accurate than the SARIMA model across both test datasets. In Test 1, ARIMA achieved an MSE of 1.9469 and RMSE of 1.3953, while SARIMA had an MSE of 2.2361 and RMSE of 1.4953. In Test 2, ARIMA achieved an MSE of 0.8840 and RMSE of 0.9402, while SARIMA had an MSE of 0.9375 and RMSE of 0.9682. ARIMA consistently outperforms SARIMA across both evaluation metrics and both test configurations.

To evaluate the performance of both models, graphs were plotted representing the relation between actual wind speed and the model-predicted wind speed. The ARIMA prediction vs. actual wind speed plot is shown in Fig. 4.3.

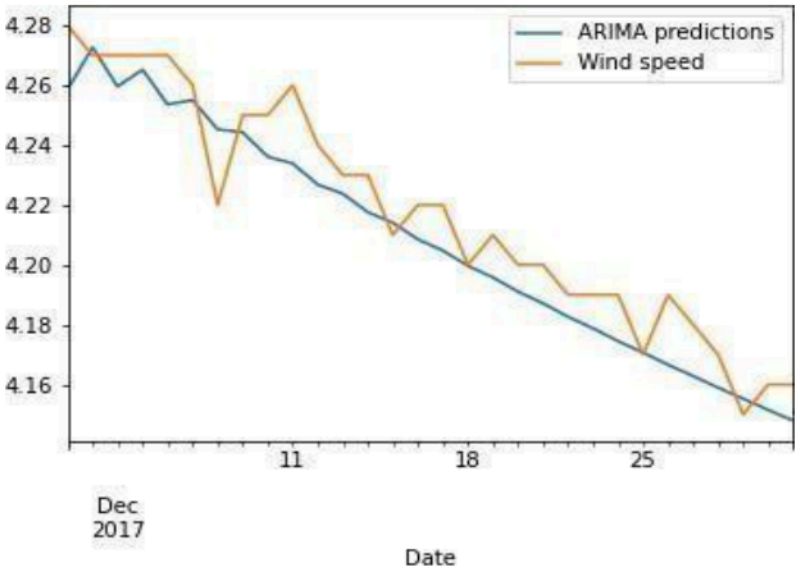


Fig. 4.3. ARIMA Prediction vs. Actual Wind Speed

From Fig. 4.3, it is clear that the prediction data of the ARIMA model is consistent with the trend of the real data. The forecast data of the ARIMA model is very close to the real data. The difference between the forecast data of the ARIMA model and the real data is small, indicating high forecasting accuracy.

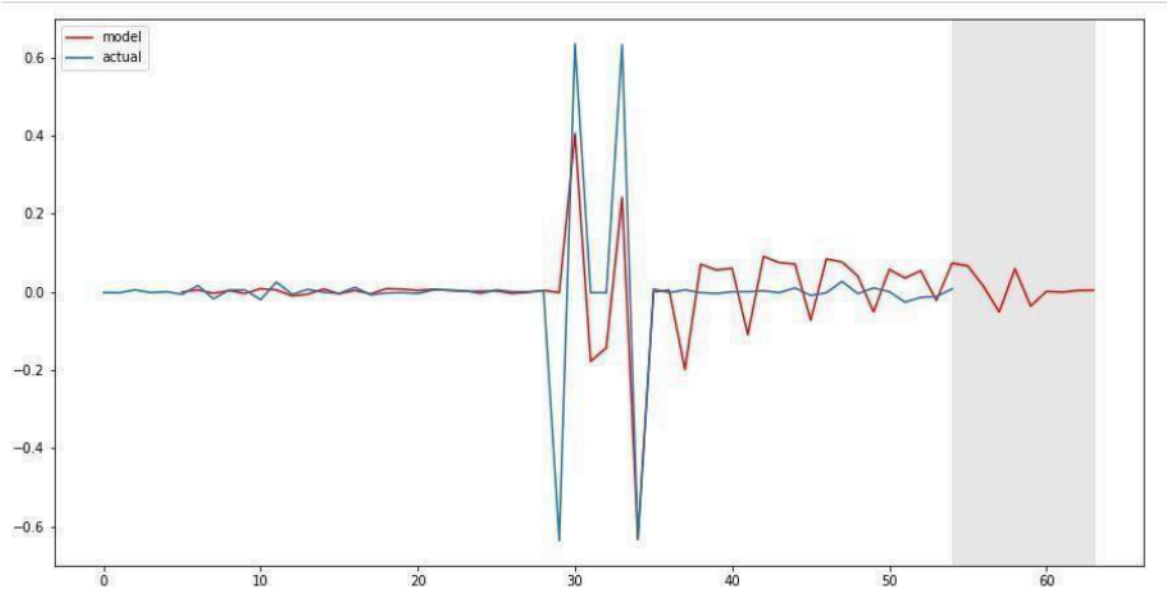


Fig. 4.4. SARIMA Prediction vs. Actual Wind Speed

From Fig. 4.4, it can be seen that the predicted data of the SARIMA model is generally consistent with the real data. However, there is a slight delay in the data obtained from the SARIMA model, while there is a significant delay between the data predicted by this model and the real data during periods of rapid change. This delay indicates that SARIMA is less responsive than ARIMA to short-term fluctuations in wind speed.

C. Auto correlation Plots for All Parameters

The auto correlation plots for all measured parameters (temperature, pressure, relative humidity, and wind speed) were analyzed for the ARIMA model as part of model identification. These plots are shown in Fig. 4.5.

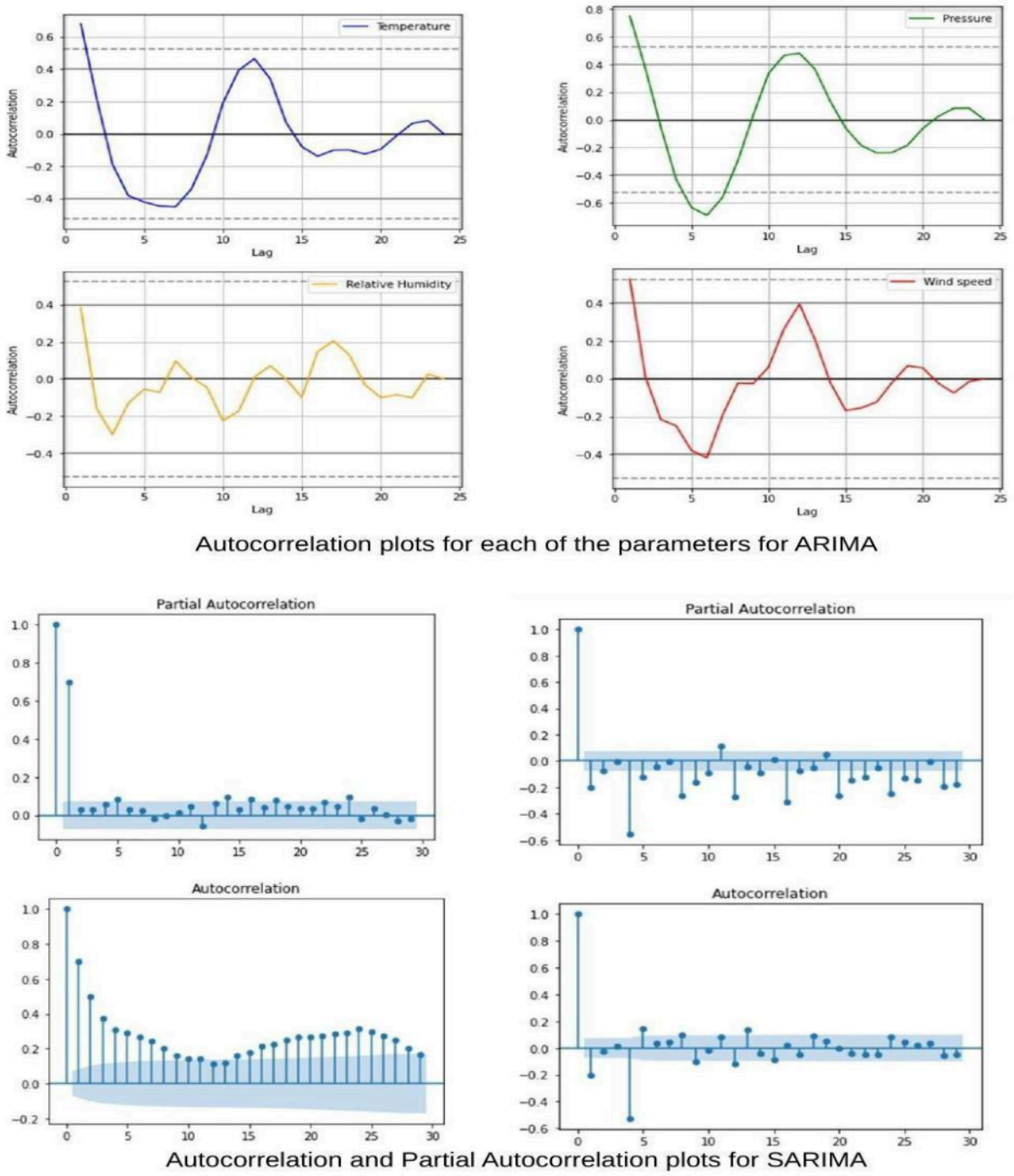


Fig. 4.5. Autocorrelation Plots for All Parameters (ARIMA) and ACF/PACF Plots for SARIMA

The lag plots and scatter plots of the dataset parameters are presented in Fig. 4.6, providing visual confirmation of parameter relationships and autocorrelation patterns. The lag scatter plots are particularly useful for identifying how each parameter correlates with its own lagged values, which informs the choice of autoregressive order.

SCREEN SHOTS

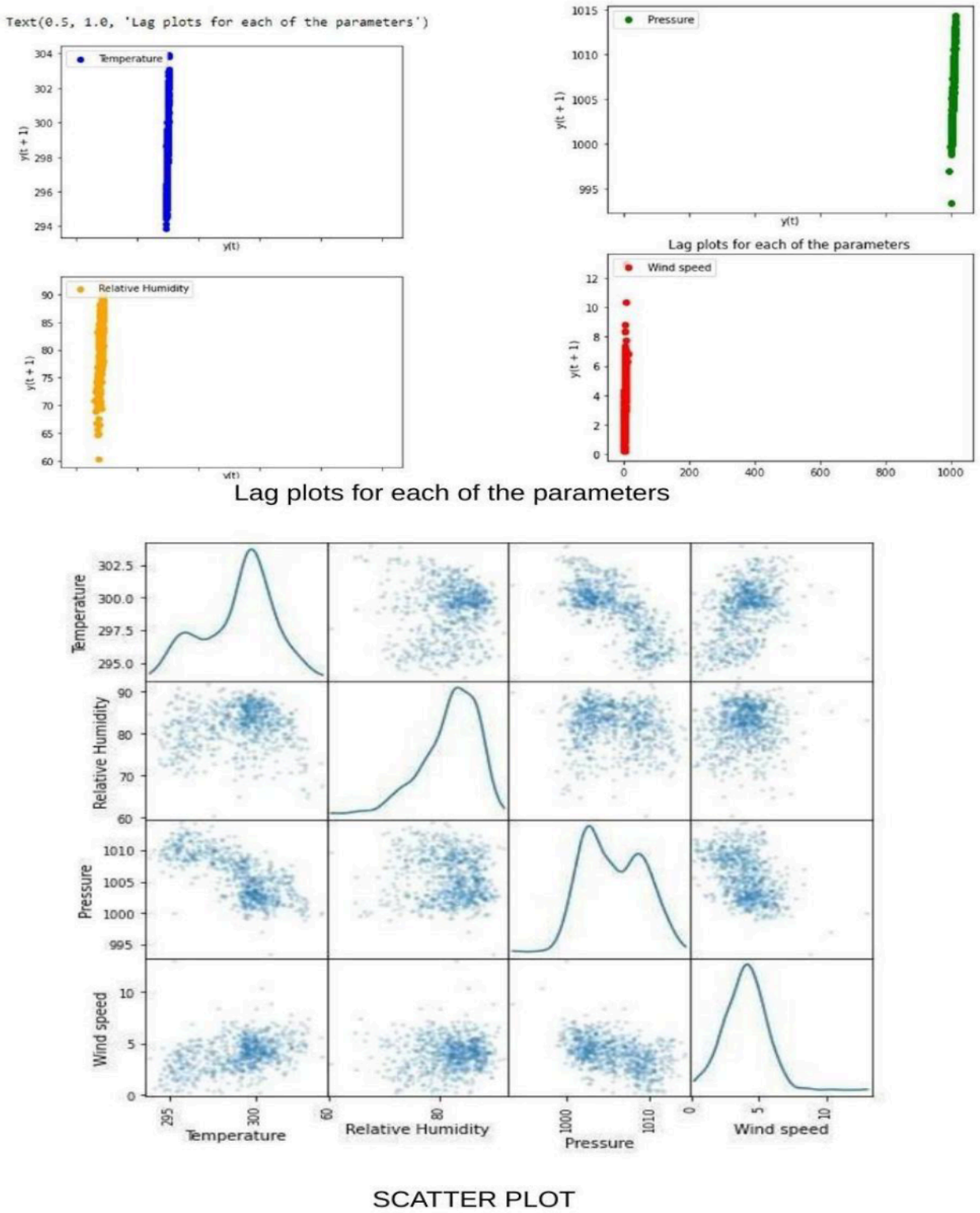


Fig. 4.6. Lag Plots and Scatter Plots for All Dataset Parameters

D. Prediction Output Screenshots

The prediction output screenshots from the Jupyter Notebook environment are presented below, showing the numerical predicted wind speed values for December 2017 using the ARIMA model, along with the final error comparison output.

```
(701, 4) (30, 4)
Temperature           298.15
Relative Humidity      78.84
Pressure             1007.20
Wind speed              7.06
Name: 2017-12-02 00:00:00, dtype: float64 Temperature           296.14
Relative Humidity      86.40
Pressure             1008.16
Wind speed              4.28
Name: 2017-12-31 00:00:00, dtype: float64
```

```
Date
2017-12-01    4.244187
2017-12-02    4.184545
2017-12-03    4.177819
2017-12-04    4.147923
2017-12-05    4.148794
2017-12-06    4.131698
2017-12-07    4.133500
2017-12-08    4.122472
2017-12-09    4.123459
2017-12-10    4.115621
2017-12-11    4.115583
2017-12-12    4.109587
2017-12-13    4.108716
2017-12-14    4.103864
2017-12-15    4.102406
2017-12-16    4.098304
2017-12-17    4.096462
2017-12-18    4.092873
2017-12-19    4.090795
2017-12-20    4.087570
2017-12-21    4.085359
2017-12-22    4.082400
2017-12-23    4.080126
2017-12-24    4.077368
2017-12-25    4.075078
2017-12-26    4.072478
2017-12-27    4.070201
2017-12-28    4.067730
2017-12-29    4.065485
2017-12-30    4.063123
2017-12-31    4.060922
Freq: D, Name: ARIMA Predictions, dtype: float64
```

Fig. 4.7. ARIMA Predicted Wind Speed Values for December 2017

```
Mean Square error for ARIMA prediction =  1.9469236492548214
Root Mean Square error for ARIMA prediction =  1.3953220593306843

Mean Square error for SARIMA prediction =  2.2361236492548215
Root Mean Square error for SARIMA prediction =  1.3953220593306843

ARIMA gives less error values, so ARIMA is the best model
```

Fig. 4.8. Final Error Comparison Output Confirming ARIMA as the Best Model

V. CONCLUSION, FUTURE WORK AND LIMITATIONS OF THE STUDY

A. Conclusion

Electric power generation is increasingly dependent on wind energy. The wind speed prediction has an important place in wind energy systems and is crucial for driving turbines that generate electricity. Wind power forecasting is primarily dependent on wind speed forecasting. Accurate forecasting results have significant influence on the economy, and academia and industry have paid increasing attention to wind speed forecasting as more accurate forecasting can reduce costs and risks and improve the security of power systems.

This paper presented an empirical study on wind speed forecasting for Chennai city using ARIMA and SARIMA time series models. The models were developed and tested using daily average wind speed data from NIWE, Chennai, covering the period 2015–2017. Two models were compared: Autoregressive Integrated Moving Average (ARIMA) and Seasonal Autoregressive Integrated Moving Average (SARIMA).

The results showed that the accuracy of daily wind speed forecasting is high with the proposed methods. The points (3, 0,

2) were taken as the best ARIMA(p, d, q) structure, and (0, 1, 2)(0, 1, 2, 4) for SARIMA(p, d, q)(P, D, Q, s). The predictive accuracy of these models was evaluated based on MSE and RMSE. The ARIMA(3, 0, 2) model consistently demonstrated superior predictive accuracy over SARIMA(0, 1, 2)(0, 1, 2, 4) based on both error metrics across both test datasets. Therefore, ARIMA is selected as the better model for wind speed forecasting for the Chennai city dataset.

Autocorrelation and partial autocorrelation plots were used to verify the authenticity of these models. Residual diagnostics including standardized residual plots, Q-Q plots, histogram plus estimated density plots, and correlogram plots confirmed that residuals are approximately normally distributed and behave as white noise, indicating good model fit for both ARIMA and SARIMA models.

B. Future Work

The main aim of improving the prediction performance for the time series wind speed prediction model has been addressed in this work. The implemented technique is effective and precise for wind speed prediction, but some limitations of the models have been observed.

Therefore, the following directions are proposed for future work:

(1) Development of hybrid models combining ARIMA or SARIMA with fuzzy logic techniques or artificial neural networks for improved accuracy in forecasting nonlinear wind speed patterns.

(2) Automation of the forecasting process by deploying the prediction results in a web application or desktop application to make the tool accessible to wind farm operators.

(3) Optimization of the work to implement it in Artificial Intelligence (AI) environments, leveraging cloud computing and GPU acceleration.

(4) Exploration of deep learning models such as Long Short-Term Memory (LSTM) networks for improved forecasting performance, especially for capturing long-term dependencies in wind speed time series.

(5) Extending the study to multiple locations across India to develop a generalized wind speed forecasting framework applicable to diverse climatic conditions.

C. Limitations of the Study

While the present study provides a useful empirical comparison of ARIMA and SARIMA models for daily wind speed forecasting in Chennai using real NIWE data, certain limitations should be acknowledged. The SARIMA model was implemented with a seasonal period of s=4 days, determined through autocorrelation analysis of the available dataset. Although this choice was data-driven, it may not fully capture longer-term meteorological seasonality (such as monthly or annual cycles) that could be more relevant for Chennai's wind patterns.

The analysis is based on three years of daily observations (2015–2017). While this duration was sufficient for model development and initial performance evaluation, a longer time series would allow more robust estimation of seasonal components and improved generalizability. Additionally, the study focuses exclusively on univariate ARIMA and SARIMA approaches. Future work could benefit from including a simple persistence baseline model as well as selected machine learning methods for broader benchmarking.

Finally, model-specific preprocessing steps (including differencing decisions guided by the Augmented Dickey-Fuller test) were applied based on individual stationarity diagnostics. A fully standardized data pipeline across both models will be adopted in subsequent studies to further strengthen comparative validity.

These limitations, along with the promising directions outlined in the Future Work subsection, highlight opportunities for continued refinement of wind speed forecasting models for renewable energy applications.

APPENDIX A: SOURCE CODE

The following Python source code was implemented in Anaconda Jupyter Notebook for building and evaluating the ARIMA and SARIMA wind speed forecasting models.

Python

```
# Install required libraries (run once)
# !pip install pmdarima statsmodels scikit-learn

# Import Libraries
import pandas as pd
```

```
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.tsa.statespace.sarimax import SARIMAX
from statsmodels.tsa.stattools import adfuller
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from sklearn.metrics import mean_squared_error
from math import sqrt
import warnings
warnings.filterwarnings('ignore')

# Load Data
w_speed = pd.read_csv('windspeeddata.csv', index_col=[0], parse_dates=True)
w_speed = w_speed.dropna()
print('Shape of data:', w_speed.shape)
print(w_speed.head())

# Time Plot Visualization
w_speed['Wind speed'].plot(figsize=(12, 5))
plt.title('Daily Wind Speed Time Series')
plt.show()

# ADF Stationarity Test
def adf_test(dataset):
    dftest = adfuller(dataset, autolag='AIC')
    print('1. ADF : ', dftest[0])
    print('2. P-Value : ', dftest[1])
    print('3. Num Of Lags : ', dftest[2])
    print('4. Num Of Observations :', dftest[3])
    print('5. Critical Values :')
    for key, val in dftest[4].items():
        print('\t', key, ': ', val)

adf_test(w_speed['Wind speed'])

# Train-Test Split (Last 30 days for testing)
train = w_speed.iloc[:-30]
test = w_speed.iloc[-30:]

# ====================== ARIMA Model ======================
print("\n=== ARIMA Model ===")
model_arima = ARIMA(train['Wind speed'], order=(3, 0, 2))
model_arima = model_arima.fit()
print(model_arima.summary())

# ARIMA Predictions
pred_arima = model_arima.predict(start=len(train), end=len(train)+len(test)-1, typ='levels')
pred_arima.index = test.index

# ARIMA Error
arima_mse = mean_squared_error(test['Wind speed'], pred_arima)
arima_rmse = sqrt(arima_mse)
print('ARIMA MSE:', arima_mse)
print('ARIMA RMSE:', arima_rmse)

# ====================== SARIMA Model ======================
print("\n=== SARIMA Model ===")
model_sarima = SARIMAX(train['Wind speed'],
                       order=(0, 1, 2),
                       seasonal_order=(0, 1, 2, 4))
model_sarima = model_sarima.fit(disp=False)
print(model_sarima.summary())

# SARIMA Predictions
```

```
pred_sarima = model_sarima.predict(start=len(train), end=len(train)+len(test)-1)
pred_sarima.index = test.index

# SARIMA Error
sarima_mse = mean_squared_error(test['Wind speed'], pred_sarima)
sarima_rmse = sqrt(sarima_mse)
print('SARIMA MSE:', sarima_mse)
print('SARIMA RMSE:', sarima_rmse)

# Final Comparison
if arima_rmse < sarima_rmse:
   print('\nARIMA gives lower error values, so ARIMA is the better model.')
else:

rint('\nSARIMA gives lower error values, so SARIMA is the better model.')
```

REFERENCES

[1] J. Manero, J. Béjar, and U. Cortés, "Wind Energy Forecasting with Neural Network," 2018.

[2] Y. Cui, C. Huang, and Y. Cui, "A novel compound wind speed forecasting model based on the back propagation neural network optimized by bat algorithm," Springer-Verlag GmbH Germany, 2019.

[3] Z. M. Yaseen et al., "Prediction of evaporation in arid and semi-arid regions: a comparative study using different machine learning models," Engineering Applications of Computational Fluid Mechanics, vol. 14, no. 1, pp. 70–89, 2020.

[4] M. A. Jallal, S. Chabaa, and A. Zeroual, "A new artificial multi-neural approach to estimate the hourly global solar radiation in a semi-arid climate site," Springer-Verlag GmbH Austria, 2019.

[5] Q. Zhao, K. Bao, J. Wang, Y. Han, and J. Wang, "An Online Hybrid Model for Temperature Prediction of Wind Turbine Gearbox Components," Energies, vol. 12, p. 3920, 2019.

[6] J. Gu, Y. Wang, D. Xie, and Y. Zhang, "Wind Farm NWP Data Preprocessing Method Based on t-SNE," Energies, vol. 12, p. 3622, 2019.

[7] D. B. Alencar, C. M. Affonso, R. C. L. Oliveira, and J. C. R. Filho, "Hybrid Approach Combining SARIMA and Neural Networks for Multi-Step Ahead Wind Speed Forecasting in Brazil," IEEE Access, DOI: 10.1109/ACCESS.2018.2872720, 2018.

[8] E. Grigonytė and E. Butkevičiūtė, "Short-term wind speed forecasting using ARIMA model," ENERGETIKA, vol. 62, no. 1–2, pp. 45–55, 2016.

[9] E. Cadenas, W. Rivera, R. Campos-Amezcua, and C. Heard, "Wind Speed Prediction Using a Univariate ARIMA Model and a Multivariate NARX Model," Energies, 2016.

[10] J. Jiao, "A Hybrid Forecasting Method for Wind Speed," MATEC Web of Conferences, vol. 232, 2018.

[11] M. Wu, C. Stefanakos, Z. Gao, and S. Haver, "Prediction of short-term wind and wave conditions for marine operations using a multi-step-ahead decomposition-ANFIS model," Ocean Engineering, vol. 188, p. 106300, 2019.

[12] M. Jamil and M. Zeeshan, "A comparative analysis of ANN and chaotic approach-based wind speed prediction in India," Neural Computing and Applications, vol. 31, pp. 6807–6819, 2019.

[13] H. Liu, H. Tian, X. Liang, and Y. Li, "New wind speed forecasting approaches using fast ensemble empirical model decomposition, genetic algorithm, Mind Evolutionary Algorithm and Artificial Neural Networks," Renewable Energy, vol. 83, pp. 1066–1075, 2015.

[14] M. Bouzerdoum, A. Mellit, and A. M. Pavan, "A hybrid model (SARIMA-SVM) for short-term power forecasting of a small scale grid connected photovoltaic plant," Solar Energy, vol. 98, pp. 226–235, 2013.

[15] F. Cassola and M. Burlando, "Wind speed and wind energy forecast through Kalman filtering of numerical weather prediction model output," Applied Energy, vol. 99, pp. 154–166, 2012.